SURFACE LANGUAGE MODELS FOR DISCOVERING TEMPORALLY ANCHORED DEFINITIONS ON THE WEB
Producing Chronologies as Answers to Definition Questions

Alejandro Figueroa
Deutsches Forschungszentrum für Künstliche Intelligenz - DFKI, Stuhlsatzenhausweg 3, D - 66123, Saarbrücken, Germany

Keywords: Web question answering, Definition questions, N-gram language models, Temporally anchored definitions.

Abstract: This work presents a data-driven definition question answering (QA) system that outputs a set of temporally anchored definitions as answers. This system builds surface language models on top of a corpus automatically acquired from Wikipedia abstracts, and ranks answer candidates in agreement with these models afterwards. Additionally, this study deals at greater length with the impact of several surface features in the ranking of temporally anchored answers.

1 INTRODUCTION

Nowadays, the systematic growth and diversification of information, published daily on the web, poses continuing and strong challenges. One of these major challenges is assisting users in finding relevant answers to natural language queries such as definition questions (e.g., "*Who is Flavius Josephus?*").

In practical terms, open-domain QA is situated at the frontier of Natural Language Processing (NLP) and modern information retrieval, being an appealing alternative to the retrieval of full-length documents. Strictly speaking, users of QA systems specify their information needs in the form of natural-language questions. This means they eliminate any artificial constraints or features that are sometimes imposed by a particular input syntax (e.g., boolean operators).

More often than not, QA systems take advantage of the fact that answers to specific questions are frequently concentrated in small fragments of text documents. This advantage helps QA systems to return brief answer strings extracted from the text collection. In a sense, it is up to the QA system to analyse the content of full-length documents and identify these small and relevant text fragments.

Essentially, definition questions are a particular category of fact–seeking queries about some topic or concept (a.k.a. *definiendum*). This type of query has become especially interesting in recent years, because of the increasing number of submissions by users to web search engines (Rose and Levinson, 2004).

More specifically, definition QA systems aim usually at finding a set of relevant and/or factual pieces of information (a.k.a. nuggets) about the *definiendum*. Nuggets are comprised of distinct kinds of pieces of information including relations with people or locations, biographical events and special attributes. Some illustrative nuggets about the *definiendum* "*Flavius Josephus*" are as follows:

also known as Yosef Ben Matityahu
Jewish
historian and apologist
born AD 37
recorded the destruction of Jerusalem in AD 70
wrote the Jewish War in AD 75
wrote Antiquities of the Jews in AD 94
In 71 became Roman citizen

Since it is necessary to provide enough context to ensure readability, definition QA systems output the corresponding set of sentences. The complexity of this task, however, is causing definition QA systems to split the problem into subtasks that independently address distinct nugget types. This work deals with one of those classes: biographical events.

Biographical events, like "born AD 37", are part of answers to definition questions encompassing chronologies of the most remarkable events or achievements related to the *definiendum*.

Figueroa A.
SURFACE LANGUAGE MODELS FOR DISCOVERING TEMPORALLY ANCHORED DEFINITIONS ON THE WEB - Producing Chronologies as Answers to Definition Questions.
DOI: 10.5220/0002769802690275
In *Proceedings of the 6th International Conference on Web Information Systems and Technology (WEBIST 2010)*, page
ISBN: 978-989-674-025-2

2 RELATED WORK

For starters, (Alonso et al., 2009) introduced the notion of the time-centred snippet as a useful way of representing documents for exploratory search and document retrieval. The core idea is profiting from sentences that carry relevant units of time (*chronons*) for building document surrogates.

Specifically, (Alonso et al., 2009) noticed that *chronons* can be incorporated into web-pages as metadata or in the form of temporal expressions. The vital aspect of these *chronons* is their relevance for presentation and for highlighting the importance of a document given a query. More precisely, they are a key factor in the construction of more descriptive snippets that include essential temporal information. In order to detect *chronons*, (Alonso et al., 2009) analysed documents for detecting temporal anchors by means of time-based linguistic tools.

As for selected sentences, (Alonso et al., 2009) only took into consideration sentences containing explicit temporal expressions. The length of these selected sentences was bound. For the purpose of ranking sentences, (Alonso et al., 2009) made allowances for the position of the temporal expression within the sentence, the number and length of the sentence, and features regarding the particular *chronon*: appearance order and its frequency in the document and within the sentence. Since (Alonso et al., 2009) applied this ranking function to a web-corpus, the features they utilised were chiefly on the surface level. Sentences are thus ranked, and the top are sorted and presented as a temporal snippet. An interesting finding of (Alonso et al., 2009) is the fact that users were concerned about the lack of time-sensitive information, that is they are keen on seeing time-sensitive information within search results. In particular, users found temporally anchored snippets as surrogates of documents very useful and the presentation of sorted temporal information interesting.

Contrarily, (Paşca, 2008) utilised temporally anchored text snippets to answer definition questions. The difference between both strategies lies in the fact that temporally anchored answers to definition questions must be biographical, and the *chronon* must be closely related to the *definiendum*, whereas temporally anchored sentences representing a document can be more diverse in nature.

Essentially, (Paşca, 2008) also focused on techniques that lack deep linguistic processing for discovering temporally anchored answers. Frequently, this type of answer must be extracted from several documents, not only because of completeness, but also as a means to increase the redundancy. In this way definition QA systems boost the probability of detecting a larger set of reliable and diverse answers that are temporally anchored, and build richer chronologies afterwards. Therefore, definition QA systems require efficient strategies that can quickly process massive collections of documents. In particular, (Paşca, 2008) processed one billion documents corresponding to the 2003 Web snapshot of Google. To be more precise, they solely used HTML tags removal, sentence detection and part-of-speech (POS) tags.

In addition, (Paşca, 2008) took advantage of a restricted set of regular expressions to detect dates: isolated year (four-digit numbers, e.g., 1977); or simple decade (e.g., 1970s); or month name and year (e.g., January 1534); or month name, day number and year (e.g., August, 1945). In order to increase the accuracy of their date matching strategy, potential dates are discarded if they are immediately followed by a noun or noun modifier, or immediately preceded by a noun. Further, four lexico-syntactic surface patterns were used for selecting answer candidates:

P_1: <Date [,|-|(|nil] [when] Snippet [,|-|)|.]>
P_2: <[StartSent] [In|On] Date [,|-|(|nil] Snippet[,|-|)|.]>
P_3: <[StartSent] Snippet [in|on] Date [EndSent]>
P_4: <[Verb] [OptionalAdverb] [in|on] Date>

As a means to avoid overmatching sentences formed by complex linguistic phenomena, they enforced nuggets on containing a verb and on carrying no pronoun. (Paşca, 2008) additionally ensured that both P_2 and P_3 match the start of the sentence, and that the nugget in P_4 contains a noun phrase. Since the aim is building a method with limited linguistic knowledge, this noun phrase was approximated by the occurrence of a noun, adjective or determiner.

Also, (Paşca, 2008) biased their ranking strategy in favour of: (a) snippets contained in a higher number of documents, and (b) snippets that carry fewer non-stop terms. By the same token, they preferred snippets that matched query words as a term to scatter query matches. Lastly, (Paşca, 2008) ranked dates in accordance with the relevance of the snippets supporting that date, and in each date, snippets are also ranked relatively to one another.

3 CORPUS ACQUISITION

Contrary to (Paşca, 2008; Alonso et al., 2009), our approach is data-driven. In short, it aims essentially at learning regularities from training sentences (positive examples) that are deemed to convey temporally anchored information about *definiendum*. More precisely, these positive examples are acquired from abstracts provided by the January 2008 snapshot of Wikipedia.

This acquisition process is motivated by the following observations: (a) sentences within abstracts are more reliable than those in the body of the article, and (b) sentences in abstracts are very likely to yield strongly related descriptions about the topic of the article (*definiendum*). Consequently, sentences carrying dates across Wikipedia abstracts are likely to yield temporal anchored definitions about their respective topic.

Specifically, Wikipedia abstracts are extracted by means of the structure supplied by the articles. With abstracts, we understand the first section provided by the document, which typically is a succinct summary of the key aspects, more important achievements and events of the corresponding topic. Also, it is worth noting that only articles fulfilling the next criterion were taken into consideration:

1. Regarding *definiendums* that their orthographical composition consists solely of numbers, letters and hyphens and periods.
2. *Definiendums* not corresponding to purpose-built pages such as lists (e.g., "*List of economists*") and categories (e.g. "*Category of magmas*").

Subsequently, selected abstracts are preprocessed as follows. Firstly, sentences are identified by means of JavaRap[1]. Secondly, sentences carrying dates are recognised by means of SupperTagger[2], and those not containing dates formed by at least one number (e.g., 1930 and March 1945) were eliminated. Thirdly, co-references are resolved. For this purpose, the replacements presented by (Keselj and Cox, 2004; Abou-Assaleh et al., 2005) were utilised. Fourthly, all instances of the *definiendum* (title of the page) are replaced with a placeholder (CONCEPT), this way the learnt models are prevented from overfitting any strong dependance between some lexical properties of the *definiendums* and their respective definitions across the training set. Also, partial matches were substituted as long as this partial match did not consider a stop-word only. Fifthly, duplicate sentences are detected by simple string matching, and all sentences that do not contain the placeholder CONCEPT were filtered out. Posteriorly, all content in parentheses was moved to the end of the sentence as a mean of preventing a distortion in the posterior learning process. Overall, a set of 2,351,708 sentences were extracted corresponding to 942,242 *definiendums* from wide-ranging topics: 2,190,630 were used for training and 161,078 for development belonging to 879,089 and 63,153 distinct *definiendums*, respectively. Some training examples include:

CONCEPT[Ocosingo, Chiapas] was given city status on 31 July 1979 .
From 1976 to 2001 , the CONCEPT[Le Studio] was used by many Canadian and international artists to record hit albums .
The CONCEPT[Le Travail Movement] lasted only 7 months , from Sept 1936 to April 1937 .
CONCEPT[Timelike Infinity] is a 1992 science fiction book by Stephen Baxter .

As a means of assessing the quality of the acquired sets, one hundred randomly selected training sentences were manually annotated. From these sentences, 17% were misleading examples. Basically, these were originated by numbers mislabelled as dates, sentence boundaries wrongly detected and wrong inferences drawn by the replacement and extraction heuristics. This acquisition process, nonetheless, provided significant accuracy, especially considering that it does not require manual annotations.

In juxtaposition, the testing set was derived from sentences taken from full web-documents, since the ultimate goal is definition QA systems operating on the web. These test sentences were obtained by submitting to a search engine (MSN Search[3]) *definiendums* that did not supply training/development sentences. A maximum of 50 hits were requested of the search engine per *definiendum*, from which only documents corresponding to web snippets containing the exact match of the *definiendum* were downloaded and processed. To be more precise, document processing consisted in removing HTML tags, and splitting them into sentences by means of openNLP[4]. Subsequently, only sentences carrying the exact match of the *definiendum* and a number were chosen, and every instance of the *definiendum* is replaced with the placeholder CONCEPT. As a result, 5,773 testing sentences were obtained belonging to 1,008 distinct *definiendums* from wide-ranging topics.

(+)On October 31 , 1948 , the CONCEPT[American Vecturist Association] was formed in New York City out of interest sparked from Mr. Moore 's newsletter .
(-)Posted by: CONCEPT[Chuck Moore] — April 19, 2007 8:35 AM

[1] http://wing.comp.nus.edu.sg/∼qiu/NLPTools/

[2] http://sourceforge.net/projects/supersensetag/

[3] http://www.live.com

[4] http://opennlp.sourceforge.net

(-)Starting in 1997 , the Union began to work with non-student instructional staff to join CONCEPT[CUPE 3902].
(+)Beginning in July of 2001 , Dr. CONCEPT[Yuri Pichugin] joined the CI team as Director of Research .

As shown in the illustrative examples, these 5,773 testing sentences were manually labelled as positive or negative. As an outcome, this manual annotation provided 1,149 positive and 4,624 negative samples respectively. Intentionally, we opted out of forcing the test set to be balanced, but rather we chose to keep the distributions as they were found on the web, this way experimental scenarios are kept as realistic as possible. Certainly, temporally anchored definitions are much fewer in number than their counterparts.

To sum up, reliable sets of positive training and development sentences were automatically acquired from Wikipedia abstracts. In each sentence, the occurrence of dates was validated by means of linguistic processing. Conversely, the testing set was obtained from the web, and numbers are interpreted as the indicator of potential dates. This key difference is due chiefly to the fact that models were trained off-line, and linguistic tools were utilised for increasing the reliability of the positive training set, and consequently, of the models. On the other hand, testing sets are assessed in real time. For this reason, dates are discriminated from numbers on the grounds of the similarity between their respective contexts and the contexts learnt by the models.

A final remark is due to on-line resources that yield temporally anchored descriptions. For instance: www.theday2day.com, www.thisdaythatyear.com, www.worldofquotes.com. We also acquired 519,240 positive examples from eight different on-line resources. We realised, however, that there are two aspects that make them less attractive: (1) the kinds of descriptions they cover is narrow, mainly births and deaths of people; and (2) the *definiendum* must be manually identified. For these two reasons, these sentences were not taken into account in this work.

4 N-GRAM LANGUAGE MODELS

Since our corpus acquisition procedure produces two sets (development and training) consisting solely of positive examples, only strategies that can learn from one-class are considered. In language models, a test sentence $S = w_1 \ldots w_l$ is scored in concert with the probability $P(S)$ of its sequence of words. This probability is usually decomposed into the multiplication of the likelihood of smaller sequences of n-words (Goodman, 2001; Figueroa and Atkinson, 2009):

$$P(S) \approx \prod_{i=1}^{l} P(w_i | w_{i-n+1} \ldots w_{i-1})$$

Where l denotes the length of the test sentence S. Typically, the length of the sentence fragments n ranges between one and five. Accordingly, $P(w_i | w_{i-n+1} \ldots w_{i-1})$ is the probability of seeing the word w_i after the fragment $w_{i-n+1} \ldots w_{i-1}$. These probabilities are normally approximated by means of the *Maximum Likelihood Estimate*:

$$P(w_i | w_{i-n+1} \ldots w_{i-1}) \approx \frac{count(w_{i-n+1} \ldots w_i)}{count(w_{i-n+1} \ldots w_{i-1})}$$

In order to deal with unknown words, when ranking test sentences, a dummy token was arbitrarily added to the model. This token was associated with a frequency value of one, then, unigrams probabilities were defined by the next formula:

$$P(w_i) \approx \frac{count(w_i)}{\sum_{\forall w} count(w_i)}$$

Subsequently, the obtained n-gram language model is smoothed by interpolating with shorter n-grams (Goodman, 2001) as follows:

$$P_{inter}(w_i | w_{i-n+1} \ldots w_{i-1}) = \lambda_n P(w_i | w_{i-n+1} \ldots w_{i-1}) +$$

$$(1 - \lambda_n) P_{inter}(w_i | w_{i-n+2} \ldots w_{i-1})$$

Lastly, the rank of a test sentence S is calculated as $Rank(S) = log(P(S))$ as a means of preventing arithmetically underflowing when dealing with long sentences. As for the smoothing parameters, their optimal values were estimated by means of the *Expectation Maximization* (EM) algorithm. In essence, this algorithm is run for each interpolation level, that is λ_5 is computed interpolating pentagrams and tetragrams, while λ_4 tetragrams and trigrams, and so on. Accordingly, the first row in table 1 underlines all parameters values for the proposed models. For the sake of clarity, the remaining two rows will be discussed later.

Table 1: Lambda Estimates.

	λ_2	λ_3	λ_4	λ_5
Language Models	0.91	0.41	0.005	0.0001
Language Models+ Google N-grams	0.80	0.26	0.014	0.0001
Language Models +CD	0.91	0.43	0.078	0.0001

5 EXPERIMENTS AND RESULTS

Our language models were tried on the testing set acquired in section 3. A baseline was implemented that assigns a random score to each sentence, and we additionally studied the impact of some of the features of (Paşca, 2008; Alonso et al., 2009) in the ranking. For the sake of simplicity, from now on, the presented system is called `ChronosDefQA`.

As to evaluation metrics, we made use of precision at k. This measurement is the ratio of sentences that are actual definitions between the first k positions of the ranking (per *definiendum*). In our experiments, we made allowances solely for the top five positions as they showed to be enough to draw a clear distinction between different approaches.

Table 2: Results achieved by the baseline and different configurations of `ChronosDefQA` (average precision at k).

	1	2	3	4	5
			`Baseline`		
random ranking	0.154	0.123	0.085	0.080	0.098
			`(1)=ChronosDefQA`		
unigrams	0.166	0.127	0.079	0.061	0.073
bigrams	0.179	0.143	0.088	0.079	0.090
trigrams	0.181	0.138	0.092	0.095	0.107
tetragrams	0.184	0.136	0.092	0.079	0.088
pentagrams	0.181	0.143	0.092	0.086	0.097
			`ChronosDefQA+Google N-grams (all)`		
unigrams	0.124	0.110	0.070	0.056	0.067
bigrams	0.123	0.111	0.069	0.075	0.092
trigrams	0.129	0.109	0.072	0.067	0.081
tetragrams	0.131	0.115	0.081	0.068	0.083
pentagrams	0.137	0.118	0.072	0.071	0.086
			`ChronosDefQA+Google N-grams (ngram)`		
unigrams	0.150	0.121	0.080	0.081	0.096
bigrams	0.148	0.122	0.071	0.077	0.093
trigrams	0.151	0.123	0.073	0.067	0.081
tetragrams	0.146	0.122	0.071	0.075	0.091
pentagrams	0.149	0.125	0.060	0.084	0.101

Table 2 stresses the achievements obtained in terms of average precision at k by the baseline and different configurations of `ChronosDefQA`. These configurations varied the level of the n-gram language models ($n = 1 \dots 5$). In light of the results, it can be concluded:

1. In general, `ChronosDefQA` outperforms our baseline in the top-three ranked positions. This suggests that our corpus in conjunction with n-gram language models enhance the performance.

2. More interestingly, trigram language models produced better results consistently, though higher level models were also taken into consideration in this evaluation. This outcome is also corroborated later by other variations of `ChronosDefQA` (see tables 3 and 4).

3. The reason to prefer language models for ranking answer candidates is having only positive samples. Thus, `ChronosDefQA` knows n-gram distributions in temporally anchored definitions, while at the same time, it lacks of data about distributions of n-grams in general text (sentences) containing numbers. As a means of inferring this information, avoiding the manual annotation of a set of sentences of about the same size than the positive set, negative (general text) language models were deduced from Google N-grams.

First, the respective frequencies of the n-grams in the positive models were taken from Google N-grams. Only these n-grams were considered in order to prevent inducing a bias in the negative models. Second, language models are built on top of these frequencies. Third, as to interpolation parameters, since we do not have annotated negative sentences, the positive development set was utilised for their tuning. Accordingly, the second row in table 1 shows the obtained values.

Two distinct alternatives were attempted. The first one, signalled by "`(all)`" in table 2, independently ranks answer candidates with both language models, and divide both scores afterwards. The second, indicated by "`(ngram)`" in table 2, divides each $P_{inter}(w_i|w_{i-n+1} \dots w_{i-1})$, and outputs the corresponding $rank(S)$. In general, both approaches were detrimental to performance, suggesting that Google N-grams did not provide a good approximation of the negative set.

4. Besides, (Alonso et al., 2009) utilised the length of the sentence as an attribute in their ranking strategy. Equally, results `(2)` in table 3 highlights the results achieved by `ChronosDefQA`, when the score function $rank(S)$ is divided by the number of tokens in S. This resulted in an enhancement with respect to `(1)` in models that account for features of higher order than unigrams. Specifically, the precision of the top ranked sentence in the trigram model increased from 0.181 to 0.194 (7.18%), while the ranking in the case of the unigram model was inclined to worsen.

5. A key aspect of $rank(S)$ is that it assigns higher scores to sentences in agreement with their likelihood of being temporally anchored definitions. Since the ranking is forced to output five sentences, in some *definiendums*, some misleading sentences can still be incorporated into the output, despite their low scores. The reason for this is two-fold: (a) few or no reliable answers were distinguished in the test set, and (b) there was no genuine answer for a particular *definiendum*. For this reason, we experimentally check

Table 3: Results achieved by distinct configurations of `ChronosDefQA` (average precision at *k*).

	1	2	3	4	5
`(2)=(1)+Length`					
unigrams	0.156	0.108	0.073	0.076	0.091
bigrams	0.194	0.150	0.110	0.109	0.122
trigrams	0.194	0.151	0.102	0.117	0.128
tetragrams	0.188	0.151	0.104	0.094	0.103
pentagrams	0.191	0.141	0.097	0.099	0.111
`(3)=(2)+Thresholds`					
unigrams	0.156	0.108	0.073	0.076	0.091
bigrams	0.203	0.158	0.117	0.117	0.131
trigrams	0.206	0.161	0.110	0.127	0.139
tetragrams	0.197	0.158	0.110	0.102	0.112
pentagrams	0.199	0.147	0.103	0.106	0.118
`(4)=(3)+ Number Substitution`					
unigrams	0.156	0.116	0.073	0.074	0.088
bigrams	0.232	0.195	0.143	0.197	0.211
trigrams	0.230	0.206	0.250	0.266	0.275
tetragrams	0.247	0.196	0.180	0.205	0.221
pentagrams	0.249	0.229	0.129	0.137	0.161
`(5)=(3)+Number Substitution & Redundancy`					
unigrams	0.182	0.140	0.087	0.094	0.106
bigrams	0.328	0.250	0.229	0.127	0.243
trigrams	0.350	0.294	0.210	0.25	0.273
tetragrams	0.307	0.253	0.184	0.260	0.286
pentagrams	0.290	0.226	0.156	0.202	0.229

for a set of reliable thresholds to tackle this issue. Accordingly, these thresholds are: unigrams(-13.0), bigrams(-4.0) and others(-2.5). The top-five answers were cut-off at any point where an answer finished with a score smaller than the respective threshold.

Results `(3)` (table 3) emphasises the new outcomes, showing that trigram models perform the best. More precisely, this model improved the performance of its homologous in `(2)` as follows: top-one (6.18%), top-two (6.6%), top-three (7.8%), top-four (8.5%) and top-five (8.6%). Since the ranking is more likely to be trimmed at lower positions, the impact of this attribute is inclined to be stronger as long as greater values of *k* are considered. This outcome also reaffirms our first conclusion, because greater improvements are achieved when cutting at lower positions, that is when discarding candidates with low scores.

6. In order to boost the similarities between the models and test sentences, numbers across training and development sentences were replaced with a placeholder (`CD`). Consequently, new language models were trained, and accordingly, the new interpolation parameters are shown in the third row of table 1. Subsequently, test sentences are assessed by `(3)`, but making use of these new language models along with substituting numbers in test sentences with `CD`.

Results `(4)` indicate the relevance of this substitution as it considerably improves the performance for almost all models in relation to `(3)`. For instance, the trigram model finished with better precision at all levels, when contrasted to its similar in `(3)`: top-one (11.65%), top-two (27.95%), top-three, top-four and five (90+%). These outcomes reaffirm the positive contribution of our language models to the ranking. In this configuration, the pentagram model outperformed the trigram model at $k = 1$ and 2, revealing the importance of fuller syntactic structures. Since numbers were replaced by a placeholder, the probability of matching some few more complete and reliable structures increased, bringing about improvements at all levels with respect to `(3)`, being these enhancements greater than the trigram models for the first and second ranked answers.

7. Incidentally, (Paşca, 2008) clustered answers according to dates, under the assumption that dates supported by more answer candidates are more likely to be the most reliable and relevant *chronons* for a particular *definiendum*. Following this observation, `ChronosDefQA` builds an histogram of the numbers appearing across answer candidates, and removes all the numbers with a frequency of one. It additionally discards dates where at least one of their instances does not occur proceeded by *a*, *the*, *in* and *on*. Contrary to (Paşca, 2008), `ChronosDefQA` accounts for any sort of number. The higher ranked instance for each date is moved to the top of the rank by adding the score of the highest ranked answer to its score. Broadly speaking, results `(5)` reveal a marked improvement with respect to `(4)` for all configurations, and for almost all levels of precision, demonstrating the relevance of the local contextual information.

8. Previous configurations, excluding those utilising Google N-grams, bias the ranking in favour of answer candidates containing positive evidence. Following the observation of (Paşca, 2008), configuration `(6)` in table 4 underlines the outcomes achieved when removing sentences containing pronouns. Results reveal an improvement in relation to `(5)`.

9. Following the same line, `ChronosDefQA` eliminates answer candidates that contain expressions including "*in*"/"*on*"/"*the*"/"*at*" (*the*) "`CONCEPT`". Like expunging pronouns, the outcome `(7)` obtained by this removal bettered the results achieved by `(6)`.

10. One aspect that makes the language models presented in this work less attractive is their assumption about characterising an answer candidate *S* as sequences of *n* words. Many times insertions and/or deletions of words can make a genuine answer look spurious or, in the best case, less reliable. As a means of investigating the impact of insertions and deletions of words, we learnt sets of five and four ordered words

Table 4: Results obtained by extra configurations of `ChronosDefQA` (average precision at *k*).

	1	2	3	4	5
`(6)=(5)+Pronouns`					
unigrams	0.190	0.149	0.100	0.092	0.104
bigrams	0.341	0.255	0.231	0.25	0.267
trigrams	0.360	0.298	0.222	0.259	0.279
tetragrams	0.315	0.257	0.194	0.270	0.298
pentagrams	0.299	0.232	0.161	0.197	0.223
`(7)=(6)+PP`					
unigrams	0.194	0.156	0.109	0.089	0.100
bigrams	0.353	0.242	0.250	0.258	0.275
trigrams	0.376	0.302	0.220	0.270	0.288
tetragrams	0.325	0.269	0.198	0.230	0.257
pentagrams	0.308	0.236	0.172	0.174	0.193
`(8)=(7)+Four-Words Orderings`					
unigrams	0.211	0.174	0.131	0.108	0.118
bigrams	0.381	0.365	0.318	0.290	0.304
trigrams	0.385	0.368	0.359	0.308	0.326
tetragrams	0.345	0.309	0.269	0.315	0.341
pentagrams	0.318	0.283	0.215	0.243	0.265
`(9)=(7)+Five-Words Orderings`					
unigrams	0.205	0.149	0.125	0.083	0.094
bigrams	0.393	0.296	0.224	0.341	0.364
trigrams	0.415	0.330	0.222	0.400	0.424
tetragrams	0.358	0.276	0.200	0.260	0.290
pentagrams	0.328	0.249	0.180	0.195	0.217

that co-occur in windows of ten tokens across training sentences, similarly to (Figueroa, 2008). These ordered words were constrained to start with the placeholder `CONCEPT` and end with the placeholder `CD`. Some illustrative highly frequent tuples are:
<`CONCEPT`,born,in,`CD`>; <`CONCEPT`,served,as,`CD`>; <`CONCEPT`,album,released,in,`CD`>;

Answer candidates matching these tuples are reranked analogously to configuration `(5)`. Generally, tuples consisting of four words caused greater enhancement at $k = 3, 4, 5$ for all models, whereas tuples formed by five words at $k = 1$ and 2. Nonetheless, both brought about an improvement with respect to the previous configuration, signalling that the context between the *definiendum* and the potential date along with its flexible word ordering is a conspicuous attribute of temporally anchored definitions.

6 CONCLUSIONS AND FUTURE WORK

This work presented an automatic acquisition method of temporally anchored definitions from Wikipedia. These acquired definitions are then used for language models that rank temporally anchored answers to definition questions. This work also studied the effects of different attributes that have a significant impact in ranking answer candidates.

As future work, we envision the enrichment of the acquisition process with more linguistic knowledge; this way more reliable training and development sets can be obtained. By the same token, contextual word orderings and windows lengths could be learnt from the respective dependency trees, and employed at the surface level to rank test sentences afterwards.

ACKNOWLEDGEMENTS

The material used in this work is available at: http://www.inf.udec.cl/∼atkinson/biographical-EventsSets.tar.gz. Thanks to John Atkinson for aiding in publishing these data-sets.

This work was partially supported by a research grant from the German Federal Ministry of Education, Science, Research and Technology (BMBF) to the DFKI project TAKE (01IW08003) and the EC-funded project QALL-ME - FP6 IST-033860 (http://qallme.fbk.eu).

REFERENCES

Abou-Assaleh, T., Cercone, N., Doyle, J., Keselj, V., and Whidden, C. (2005). DalTREC 2005 QA System Jellyfish: Mark-and-Match Approach to Question Answering. In *Proceedings of TREC 2005*. NIST.

Alonso, O., BaezaYates, R., and Gertz, M. (2009). Effectiveness of temporal snippets. In *Workshop on Web Search Result Summarization and Presentation (WWW 2009)*.

Figueroa, A. (2008). Mining Wikipedia for Discovering Multilingual Definitions on the Web. In *4th International Conference on Semantics, Knowledge and Grid*, pages 125–132.

Figueroa, A. and Atkinson, J. (2009). Using Dependency Paths For Answering Definition Questions on The Web. In *5th International Conference on Web Information Systems and Technologies*, pages 643–650.

Goodman, J. T. (2001). A bit of progress in language modeling. *Computer Speech and Language*, 15:403–434.

Keselj, V. and Cox, A. (2004). DalTREC 2004: Question Answering Using Regular Expression Rewriting. In *Proceedings of TREC 2004*. NIST.

Paşca, M. (2008). Answering Definition Questions via Temporally-Anchored Text Snippets. In *Proceedings of IJCNLP*.

Rose, D. E. and Levinson, D. (2004). Understanding user goals in web search. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 13–19.